MPI behavior under PBS

A "*chunk*" is one node i.e. "**select=2**" requests two nodes.

$PBS_NODEFILE contains one entry per node for both "**select=<num_nodes>**" and "**select=<num_nodes>:ncpus=<num_cpus>**"

$PBS_NODEFILE contains <**num_mpi_procs**> entries per node for "**select=<num_nodes>:mpiprocs=<num_mpi_procs>**"

For IntelMPI (using mpi/impi-4.0.1.002-beta for testing), there are three ways to run an MPI job.

- First way: Use "*mpirun*" i.e.
    - mpirun -np <num_procs> <executable>

    This way prevents oversubscription in that <num_procs> cannot be greater than the number of entries in $PBS_NODEFILE. Therefore, to use this way, one needs to do "**select=<num_nodes>:mpiprocs=<num_mpi_procs**>" so that *$PBS_NODEFILE* will contain the intended maximum number of MPI processes and *num_procs <= num_nodes*num_mpi_procs*

- Second way: Use "*mpdboot*" followed by "*mpiexec*" i.e.
    - mpdboot -r sshmpi -n <num_nodes> -f $PBS_NODEFILE
    - mpiexec -n <num_procs> <executable>

    For the version of MPI tested, this approach appears not to work correctly in that "mpiexec" fails to distribute processes across nodes i.e. it puts all processes on the head node of the allocation. This used to work correctly in earlier versions of Intel MPI. We need to run more tests and file a bug report with Intel if necessary. For now, we shouldn't encourage this approach to be used.

For both the first and the second ways, the "-perhost" option appears not to change anything.

- Third way: Use "*mpirun.actual*" i.e.
    - mpirun.actual -np <num_procs> <executable>

    This way offers maximum flexibility in that it correctly recognizes the number of unique nodes in *$PBS_NODEFILE* even if there is more than one entry per node. It also distributes MPI processes across nodes. It also responds to "-*perhost*" option as a means of controlling at run time how many MPI processes one will like to have per node. I will recommend this approach since

it appears to be the easiest, most flexible and least confusing. For this approach, all one has to do is "*select=<num_nodes>*".

For MVAPICH2, only one way, using "*mpirun_rsh*", works satisfactorily i.e.

- mpirun_rsh -hostfile $PBS_NODEFILE -np <num_procs> <executable>

Both "*mpirun*" and "*mpdboot/mpiexec*" do not work because they are referencing Python2.4 that is no longer available on the system.
"*mpirun_rsh*" correctly determines the number of nodes and distributes work accordingly.

For OpenMPI, "*mpirun*" works satisfactorily i.e.

- mpirun -hostfile $PBS_NODEFILE -np <num_procs> <executable>

"*mpirun*" correctly determines the number of nodes and distributes work accordingly. It seems to determine the number of unique nodes in the *$PBS_NODEFILE* correctly even if each node has more than one entry.

N.B. - For "mpirun.actual" in Intel MPI, "mpirun_rsh" in MVAPICH2 and "mpirun" in OpenMPI, the user is not prevented from oversubscription so it is the user's responsibility to request appropriate number of nodes.